IN THE CLAIMS

Please amend the claims to the following.

1. (Currently Amended) An apparatus ~~system~~ for maintaining cache coherency ~~in a CMP~~ comprising:

an integrated circuit including

a plurality of processor cores, wherein the plurality of processor cores each include a private cache;

a shared cache to be shared by the plurality of processor cores, wherein the shared cache includes logic, in response to receiving a write request referencing a block from a requesting processor core of the plurality of processor cores and the block not being owned, to generate a first message including an invalidation part and a write-acknowledgement part, and wherein at least the invalidate part of the first message when received by a second processor core of the plurality of processor cores is to invalidate the block in the second processor core and at least the write-acknowledgement part the first message, when received by the requesting processor core, is also to act as a write acknowledgement to the requesting processor core; and

a ring to connect the plurality of processor cores and the shared cache, the ring to transmit the first message to the requesting processor core and second processor core.

2. (Canceled)

3. (Currently Amended) The apparatus ~~system~~ of claim 1 wherein the shared cache includes one or more banks, wherein the one or more cache banks is responsible for a subset of a physical address space of the system, and wherein the block is associated with a physical address of the physical address space of the system.

4. (Currently Amended) The apparatus ~~system~~ of claim 1 wherein the first message includes an InvalidateAndAcknowledge message , and wherein the shared cache is to generate the InvalidateAndAcknowledge message, further in response to the block being present in the shared cache and the second processor core being a custodian for the block.

5. (Currently Amended) The apparatus ~~system~~ of claim 1 wherein the first message includes an InvalidateAllAndAcknowledge message, and wherein the shared cache, in response to receiving the write request referencing the block from the requesting processor core of the plurality of processor cores and the block not being owned, is to generate the InvalidateAllAndAcknowledge message, further in response to the block not being present in the shared cache and none of the plurality of processor cores being a custodian for the block.

6. (Currently Amended) The apparatus ~~system~~ of claim 1 wherein the plurality of processor cores writes data through to the shared cache.

7. (Currently Amended) The apparatus ~~system~~ of claim 1 wherein the plurality of processor cores each include a merge buffer, and wherein each of the merge buffers are to coalesce multiple stores to a same block.

8. (Currently Amended) The apparatus ~~system~~ of claim 1 wherein the shared cache is to fetch a second block from a memory and generate a write acknowledge message to provide a write acknowledgement to the requesting processor core in response to receiving a second write request referencing the second block, the second block not being present in the shared cache and not being owned by any of the plurality of processor cores.

9. (Currently Amended) The apparatus ~~system~~ of claim 8 wherein the shared cache is to generate an evict message to evict a third block from an owning processor core and generate a second write acknowledge message to provide a second write acknowledgment to the requesting processor core in response to receiving a third write request referencing the third block, the third block being present in the shared cache and the owning processor core of the plurality of cores owns the third block.

10. (Currently Amended) The apparatus ~~system~~ of claim 1 wherein a bank of the shared cache is to be a home location for a non-overlapping portion of a physical address space associated with the block.

11. (Currently Amended) The apparatus ~~system~~ of claim 7 wherein each private cache of the plurality of cores are not to hold dirty data, and wherein each of the merger buffers are to hold the dirty data.

12. (Currently Amended) The apparatus ~~system~~ of claim 1 wherein the ring is a synchronous, unbuffered bidirectional ring interconnect.

13. (Currently Amended) The apparatus ~~system~~ of claim 12 wherein the first message has a fixed deterministic latency around the ring interconnect.

14. (Currently Amended) An apparatus comprising:

an integrated circuit including: a plurality of cores and a shared memory connected in a ring, the shared memory to be accessible by each of the plurality of cores, wherein each of the plurality of cores includes a private memory and a merge buffer to purge data to the shared memory, and wherein the shared memory includes receiving logic to receive, from a requesting core of the plurality of cores, a read request referencing the address, ownership logic to determine an owning processor core of the plurality of processor cores owns a block associated with the address, and eviction logic coupled to the receiving logic and the ownership logic, the eviction logic to generate an evict message referencing the ~~an~~ address and ~~associated with~~ the ~~an~~ owning processor core ~~of the plurality of cores~~ in response to the receiving logic receiving the ~~a~~ read request ~~referencing the address from a requesting core of the plurality of cores~~ and the ownership logic determining the owning processor core owns[[ing] the ~~a~~ block ~~associated with the address~~.

15. (Previously Presented) The apparatus of claim 14, wherein the ring includes a synchronous unbuffered bi-directional ring interconnect.

16. (Previously Presented) The apparatus of claim 14, wherein the shared memory is a shared cache including a plurality of blocks, and wherein the shared cache is capable of holding each of the plurality of blocks in a cache coherency state.

17. (Previously Presented) The apparatus of claim 16, wherein the cache coherency state for each of the plurality of blocks is selected from a group consisting of (1) a not present state, (2) a present and owned by a core of the plurality of cores state, (3) a present, not owned, and custodian is a core of the plurality of cores state, and (4) a present, not owned, and no custodian state.

18. (Currently Amended) A system comprising:

a processor including: a plurality of cores and a shared memory to be coupled together with an unbuffered bi-directional ring interconnect, wherein each of the plurality of cores is to be associated with a private cache memory, the shared memory is to be accessible by each of the plurality of cores, and the shared memory is to include a plurality of blocks, each of the plurality of blocks capable of being held by logic in the shared memory in a not present state;[[,]] a present and owned by a core of the plurality of cores state;[[,]] a present, not owned, and a core of the plurality of cores is a custodian state;[[,]] and a present, not owned, and no core of the plurality of cores is a custodian state; and

a system memory associated with the processor to hold elements to be stored by the shared memory.

19. (Previously Presented) The system of claim 18, wherein each of the plurality of blocks is a home location for a subset of a physical address space.

20. (Previously Presented) The system of claim 19, wherein the shared cache is to generate a first message to invalidate a requested block in all cores of the plurality of cores except for a requesting core of the plurality of cores, in response to receiving a write request referencing the requested block from the requesting core and requested block being held in the present, not owned, and no core of the plurality of cores is a custodian state.

21. (New) A method for maintain cache coherency comprising:

receiving, with a shared cache, a write request referencing a block from a requesting processor core of the plurality of processor cores on a processor, wherein the plurality of processor cores each include a private cache, and wherein the plurality of cores and the shared cache are connected by a ring interconnect;

generating a single message, with the shared cache, in response to receiving the write request;

transmitting the single message on the ring interconnect to at least a second processor core of the plurality of processor cores and to the requesting processor core;

invalidating the block in the private cache included in the second processor core in response to the second processor core receiving the single message transmitted on the ring interconnect; and

write-acknowledging the write request for the requesting processor core in response to the requesting processor core receiving the single message transmitted on the ring interconnect.

22. (New) The method of claim 21, wherein the shared cache includes one or more banks, wherein the one or more cache banks is responsible for a subset of a physical address space of a computer system including the processor, and wherein the block is associated with a physical address of the physical address space of the computer system.

23. (New) The method of claim 21 wherein the first message includes an InvalidateAndAcknowledge message , and wherein generating the InvalidateAndAcknowledge message, with the shared cache, is further in response to the block being present in the shared cache and the second processor core being a custodian for the block.

24. (New) The method of claim 21 wherein the first message includes an InvalidateAllAndAcknowledge message, and wherein generating the InvalidateAllAndAcknowledge message, with the shared cache, is further in response to the block not being present in the shared cache and none of the plurality of processor cores being a custodian for the block.

25. (New) The method of claim 21 wherein the plurality of processor cores writes data through to the shared cache.

26. (New) The method of claim 21 wherein the plurality of processor cores each include a merge buffer, and wherein each of the merge buffers are to coalesce multiple stores to a same block.

27. (New) The method of claim 21, further comprising fetching, with the shared memory, a second block from a memory and generating, with the shared memory,

a write acknowledge message to provide a write acknowledgement to the requesting processor core in response to receiving a second write request referencing the second block, the second block not being present in the shared cache and not being owned by any of the plurality of processor cores.

28. (New) The method of claim 27 further comprising generating, with the shared cache, an evict message to evict a third block from an owning processor core of the plurality of processor cores and generating a second write acknowledge message to provide a second write acknowledgment to the requesting processor core in response to receiving a third write request referencing the third block, the third block being present in the shared cache and the owning processor core of the plurality of cores owns the third block.

29. (New) The method of claim 21 wherein a bank of the shared cache is to be a home location for a non-overlapping portion of a physical address space associated with the block.

30. (New) The method of claim 26 wherein each private cache including in the plurality of cores are not to hold dirty data, and wherein each of the merger buffers are to hold the dirty data.

31. (New) The method of claim 21 wherein the ring interconnect includes a synchronous, unbuffered, bidirectional, ring interconnect.

32. (New) The method of claim 21 wherein the first message has a fixed deterministic latency around the ring interconnect.